**Image Classification Project Documentation**

**Overview**

This project involves building an image classification model using transfer learning with MobileNetV2 on the CIFAR-10 dataset. The dataset contains 60,000 labeled images in 10 classes. The project covers data loading, preprocessing, model building, training, fine-tuning, and evaluation.

**Step-by-Step Process**

**1. Import Libraries**

TensorFlow, pandas, matplotlib, numpy are imported for data manipulation, visualization, and modeling.

Keras modules for datasets, layers, and models are used.

**2. Load CIFAR-10 Dataset**

The dataset is loaded with 50,000 training images and 10,000 test images, each 32x32 pixels with 3 color channels.

Classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

**3. Data Inspection and Visualization**

Shapes of training and test sets are inspected.

Sample images are displayed with their corresponding class labels.

Labels are reshaped (flattened) for compatibility with the model.

**4. Dataset Splitting**

Training set is split into training and validation sets using 80:20 ratio (40k training, 10k validation).

Validation labels are flattened.

**5. Data Preprocessing**

Defined a preprocess function to resize images to 224x224 pixels (required input for MobileNetV2) and normalize pixel values too.

TensorFlow Dataset API creates batches and shuffles training data.

Validation and test datasets are batched.

**6. Model Creation Using Transfer Learning**

MobileNetV2 pretrained on ImageNet is loaded without the top layer.

Base model layers are initially frozen to preserve pretrained weights.

Added GlobalAveragePooling2D layer to reduce feature maps to single value per channel.

Added Dense layer with 128 units and ReLU activation.

Added output Dense layer with 10 units (for CIFAR-10 classes) and Softmax activation.

**7. Model Compilation**

Compiled with Adam optimizer.

Loss function is sparse categorical crossentropy (for integer labels).

Metrics include accuracy.

**8. Model Training**

Trained on preprocessed training dataset.

Validation performed on validation dataset.

Training accuracy and validation accuracy monitored over epochs.

**Progressive Layer Unfreezing and Fine-Tuning**

**Unfreeze Last 10 Layers**

After initial training with base model frozen, the last 10 layers of MobileNetV2 were unfrozen.

Layers before these last 10 remain frozen.

Model recompiled with a small learning rate 1×10−51×10−5.

Fine-tuned for several epochs to adjust weights more specifically for CIFAR-10 data.

**Unfreeze Last 30 Layers**

After training with last 10 layers unfrozen, further fine-tuning done by unfreezing last 30 layers.

Model again recompiled with small learning rate.

Fine-tuned the model to adapt more complex features

**Overfitting Prevention Techniques**

* **Dropout Layer**

Randomly disables some neurons during training.

Helps in generalization by preventing co-adaptation on training data.

* **EarlyStopping Callback**

Monitors validation accuracy or loss.

Stops training if no improvement in validation metrics over a set number of epochs (patience=3).

Reduces overfitting by avoiding prolonged training.

**Evaluation and Results**

Training and validation accuracy plotted to assess generalization.

Improved accuracy observed with layer fine-tuning.

Early stopping prevented overfitting.

The model saved for deployment in Streamlit.

This documentation comprehensively covers each step, analysis, and approach used in the image classification project on CIFAR-10 using MobileNetV2 transfer learning with fine-tuning and regularization techniques applied. Training accuracy of 0.9422 achieved.